### Solving analytical queries on Redshift Cluster

1. **Top 10 ATMs where most transactions are in the ’inactive’ state**

SELECT atm\_number, atm\_manufacturer, location, total\_transaction\_count, inactive\_count

FROM

(SELECT c.atm\_number, c.atm\_manufacturer, b.location,

COUNT(a.trans\_id) as total\_transaction\_count,

SUM(CASE WHEN a.atm\_status = 'Inactive' THEN 1 ELSE 0 END) inactive\_count,

DENSE\_RANK() OVER(ORDER BY inactive\_count desc) rank

FROM upgrad\_etl\_proj.fact\_atm\_trans a,

upgrad\_etl\_proj.dim\_location b, upgrad\_etl\_proj.dim\_atm c

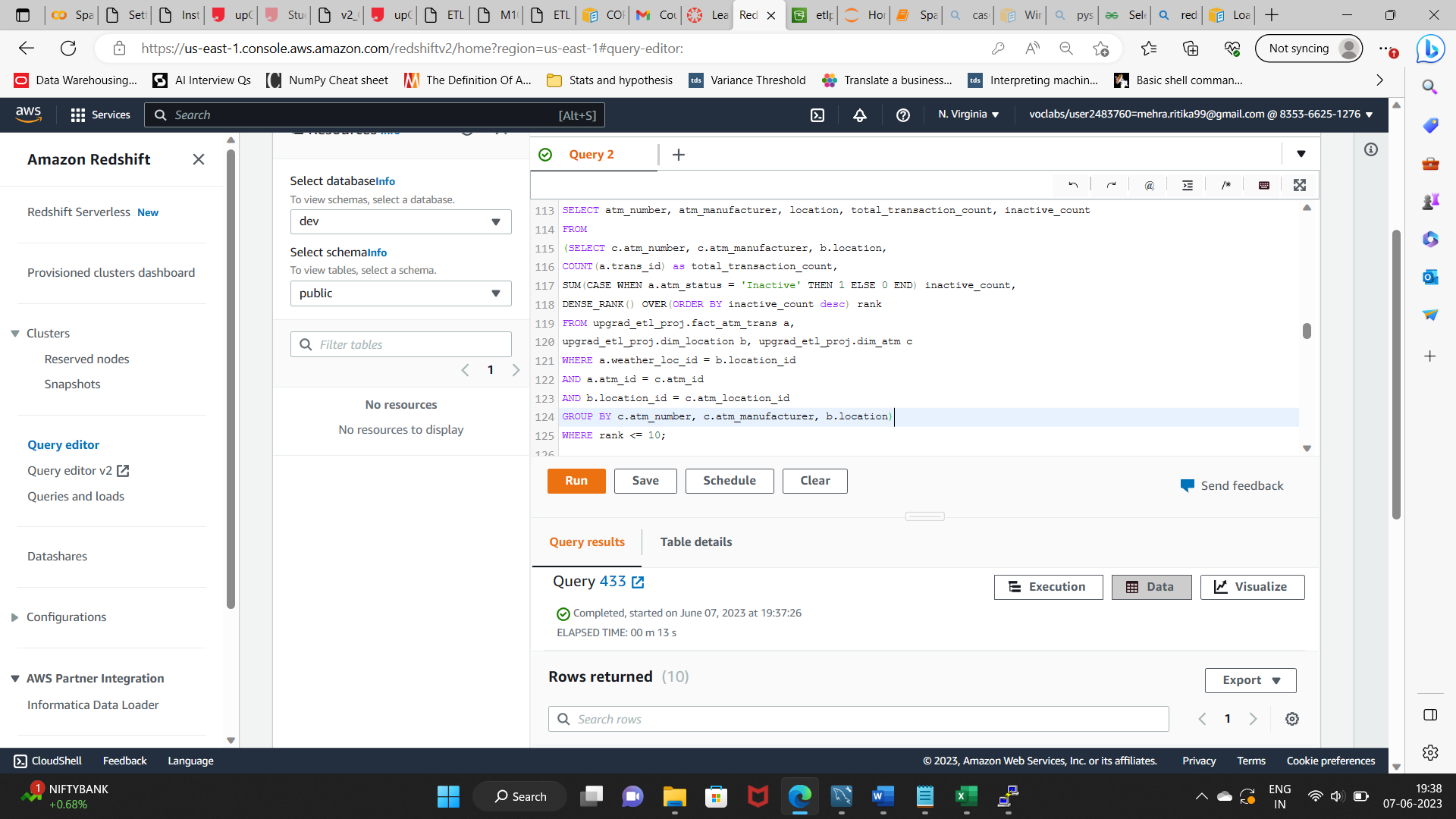
WHERE a.weather\_loc\_id = b.location\_id

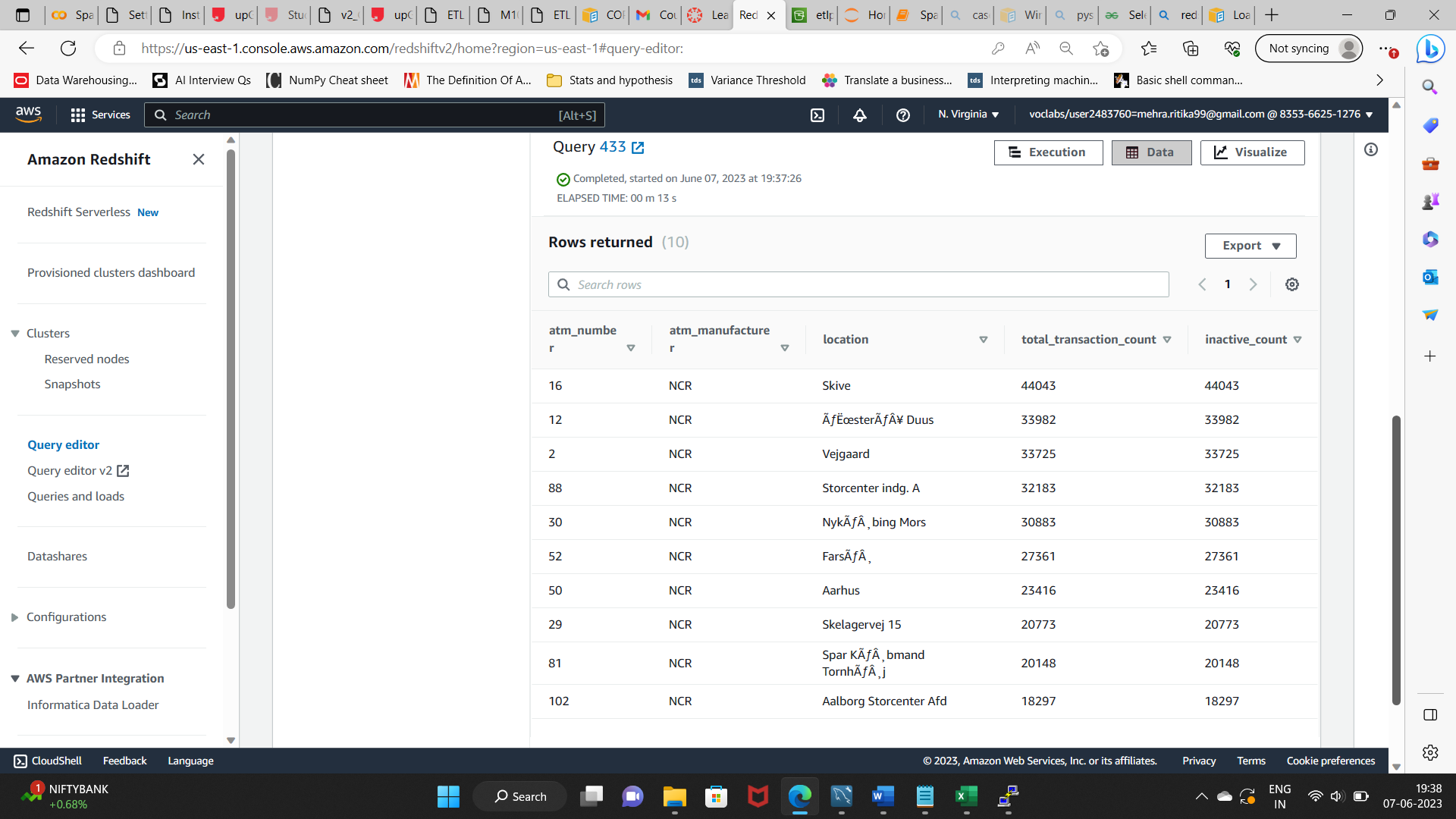
AND a.atm\_id = c.atm\_id

AND b.location\_id = c.atm\_location\_id

GROUP BY c.atm\_number, c.atm\_manufacturer, b.location)

WHERE rank <= 10;





1. **Number of ATM failures corresponding to the different weather conditions recorded at the time of the transactions**

SELECT a.weather\_main, COUNT(a.trans\_id) as total\_transaction\_count,

SUM(CASE WHEN a.atm\_status = 'Inactive' THEN 1 ELSE 0 END) inactive\_count,

CAST(ROUND(inactive\_count\*100/total\_transaction\_count::double precision,2) AS DECIMAL (8,4)) inactive\_count\_percent

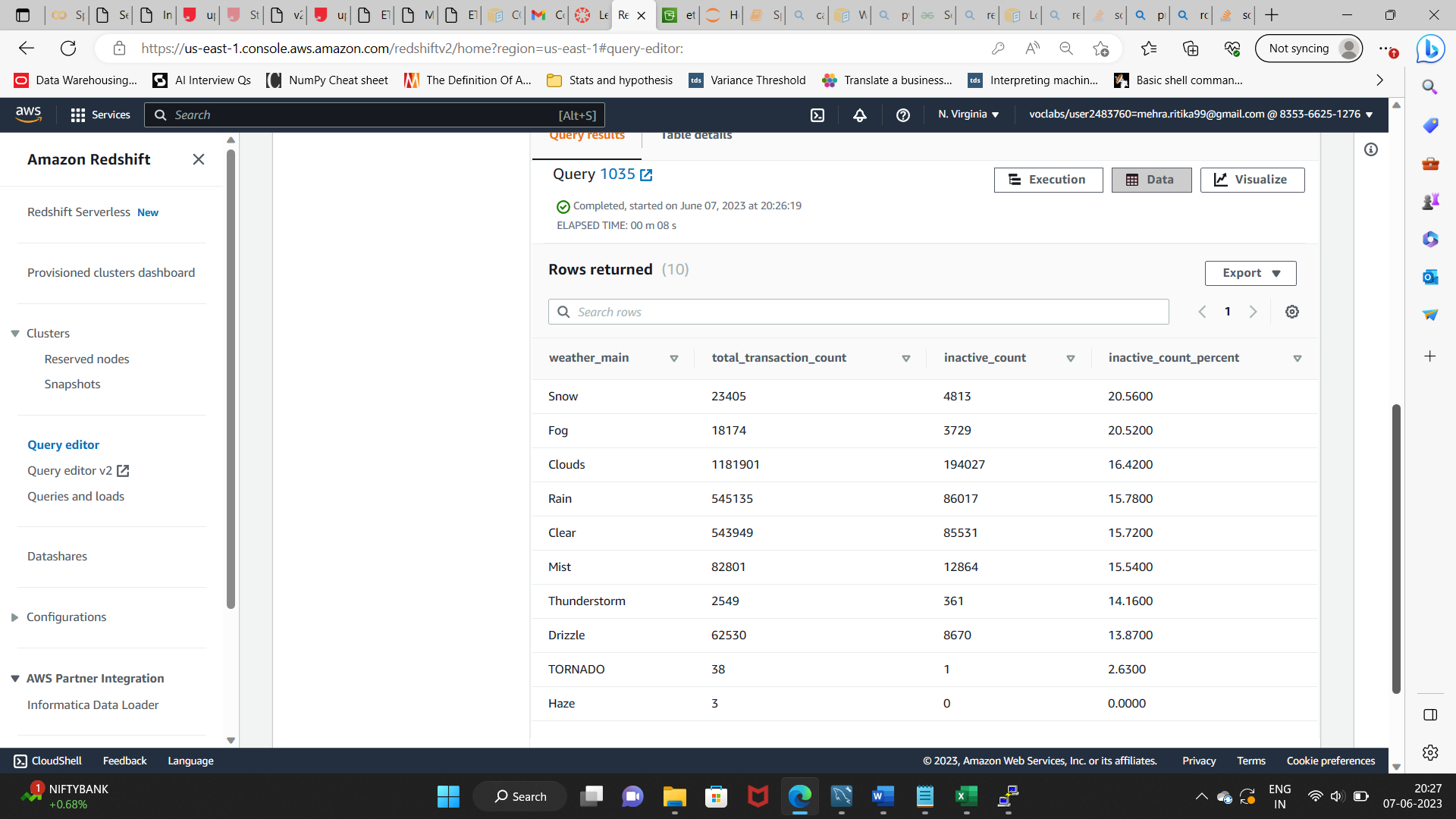
FROM upgrad\_etl\_proj.fact\_atm\_trans a

where a.weather\_main != 'Other'

GROUP BY weather\_main

ORDER BY inactive\_count\_percent desc;





1. **Top 10 ATMs with the most number of transactions throughout the year**

SELECT atm\_number, atm\_manufacturer, location, total\_transaction\_count

FROM

(SELECT c.atm\_number, c.atm\_manufacturer, b.location, COUNT(a.trans\_id) total\_transaction\_count, DENSE\_RANK() OVER(ORDER BY count(a.trans\_id) desc) rank

FROM upgrad\_etl\_proj.fact\_atm\_trans a, upgrad\_etl\_proj.dim\_location b, upgrad\_etl\_proj.dim\_atm c

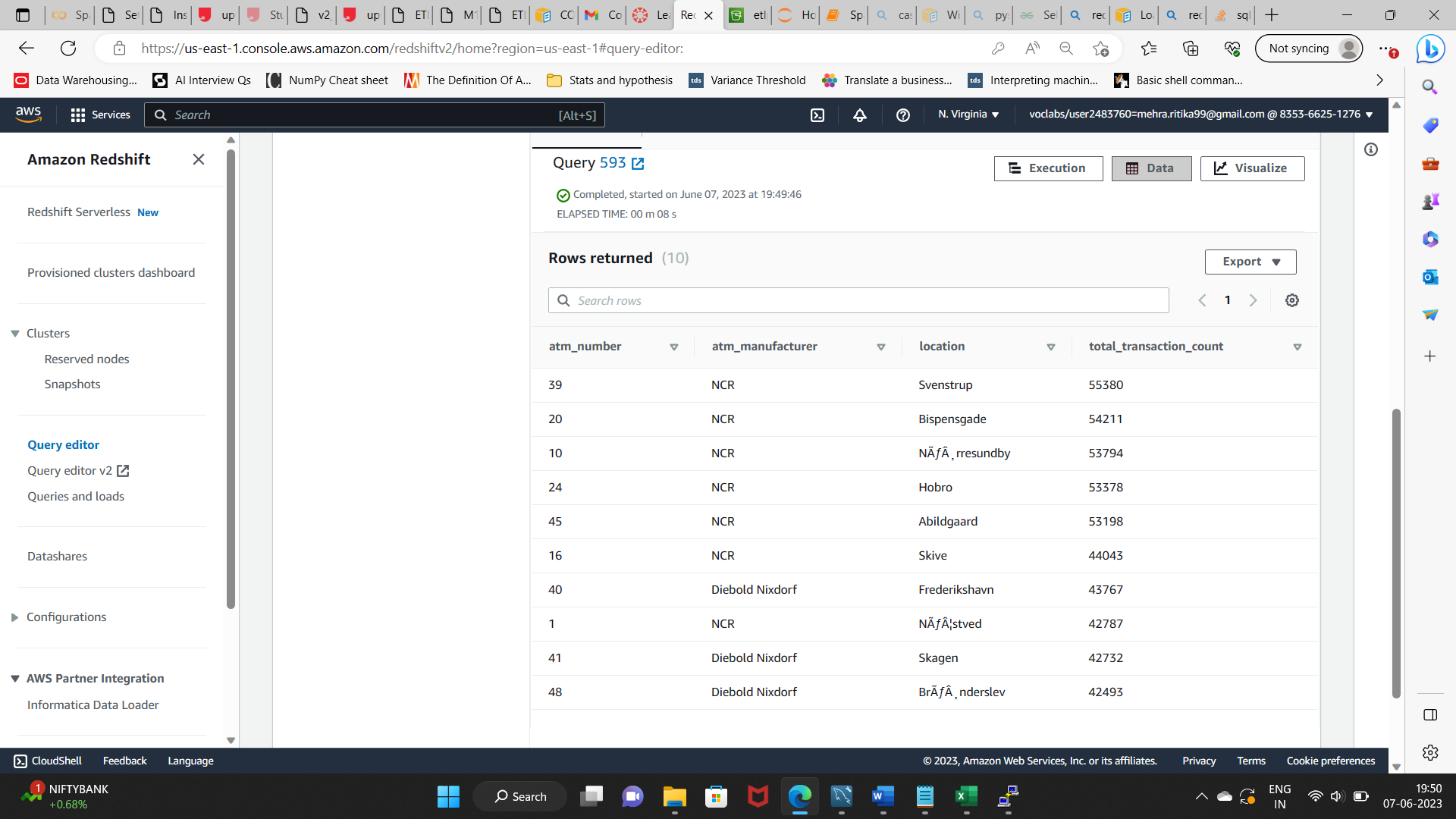
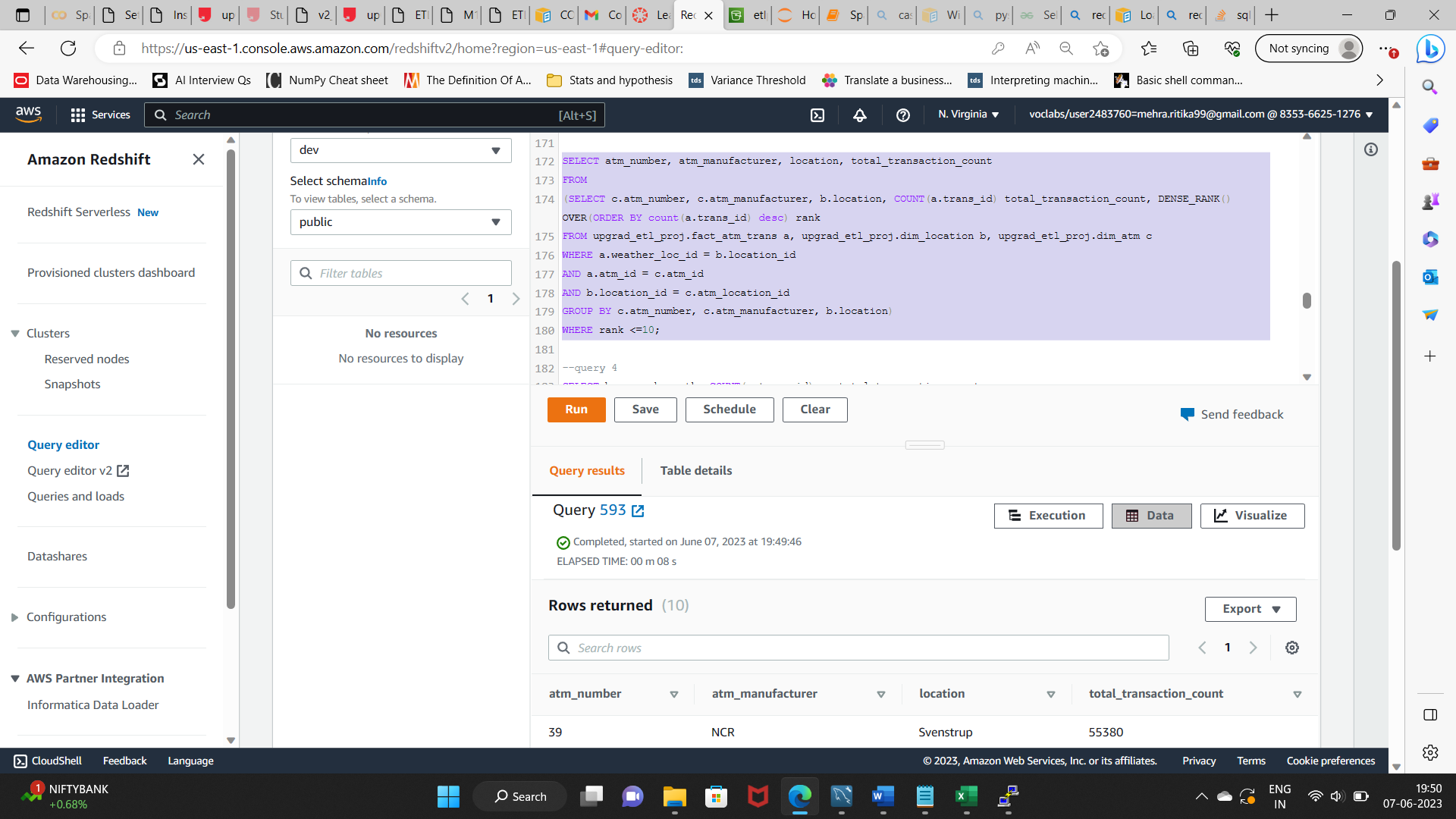
WHERE a.weather\_loc\_id = b.location\_id

AND a.atm\_id = c.atm\_id

AND b.location\_id = c.atm\_location\_id

GROUP BY c.atm\_number, c.atm\_manufacturer, b.location)

WHERE rank <=10;



1. **Number of overall ATM transactions going inactive per month for each month**

SELECT b.year, b.month, COUNT(a.trans\_id) as total\_transaction\_count,

SUM(CASE WHEN a.atm\_status = 'Inactive' THEN 1 ELSE 0 END) inactive\_count,

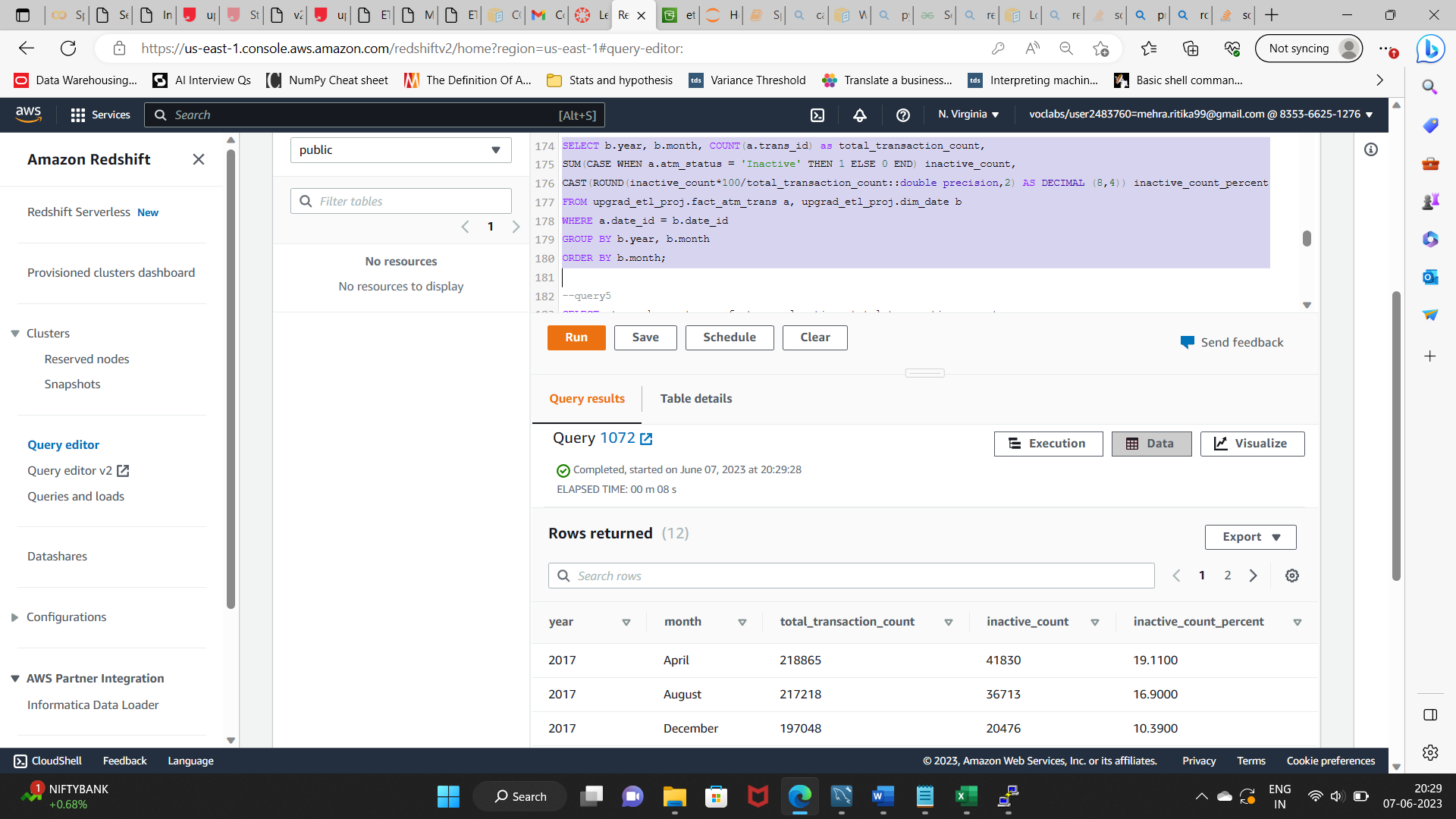
CAST(ROUND(inactive\_count\*100/total\_transaction\_count::double precision,2) AS DECIMAL (8,4)) inactive\_count\_percent

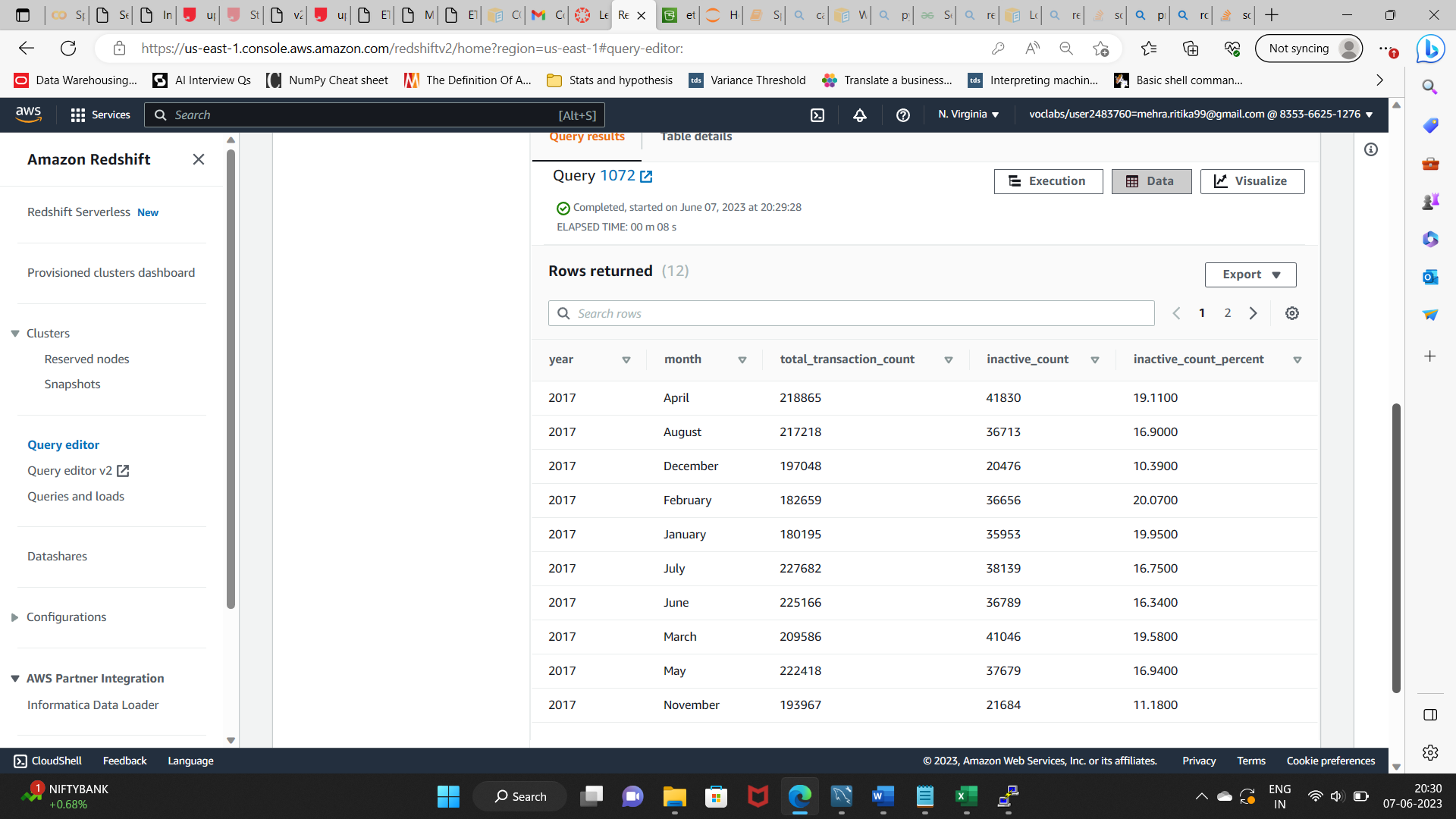
FROM upgrad\_etl\_proj.fact\_atm\_trans a, upgrad\_etl\_proj.dim\_date b

WHERE a.date\_id = b.date\_id

GROUP BY b.year, b.month

ORDER BY b.month;





1. **Top 10 ATMs with the highest total withdrawn amount throughout the year**

SELECT atm\_number, atm\_manufacturer, location, total\_transaction\_amount

FROM(SELECT c.atm\_number, c.atm\_manufacturer, b.location, SUM(a.transaction\_amount) total\_transaction\_amount, DENSE\_RANK() OVER(ORDER BY SUM(a.transaction\_amount) desc) rank

FROM upgrad\_etl\_proj.fact\_atm\_trans a, upgrad\_etl\_proj.dim\_location b, upgrad\_etl\_proj.dim\_atm c

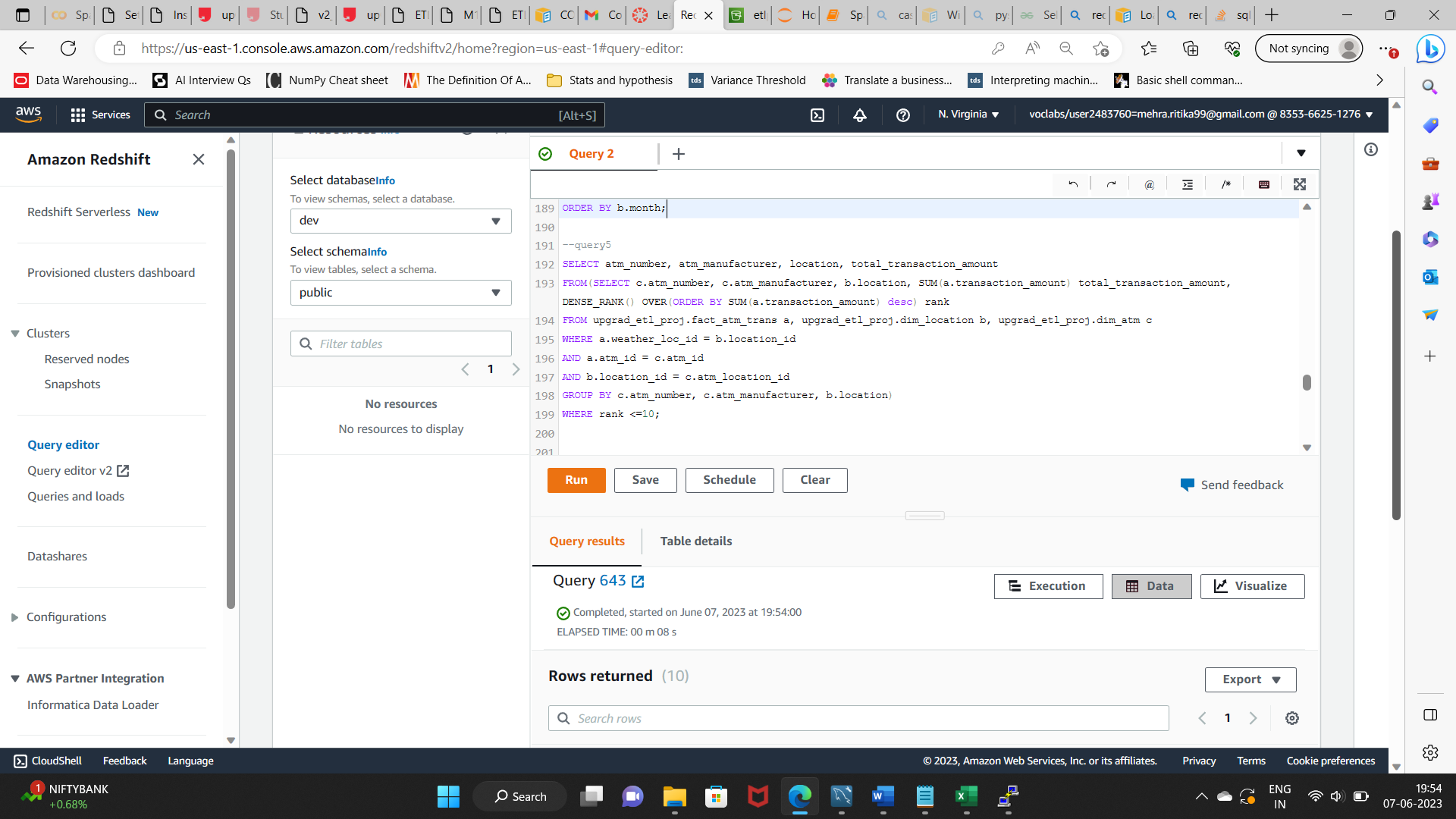
WHERE a.weather\_loc\_id = b.location\_id

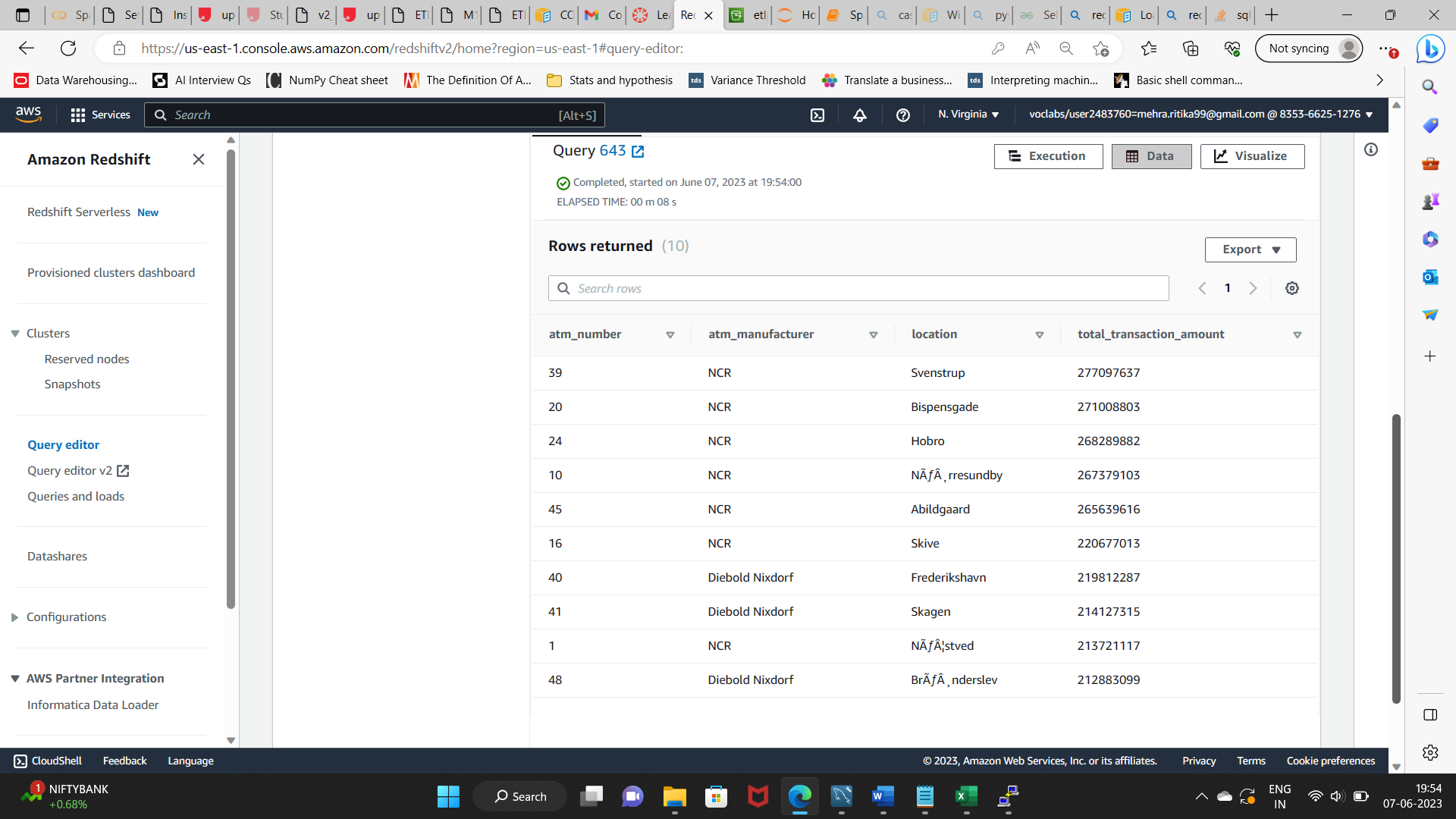
AND a.atm\_id = c.atm\_id

AND b.location\_id = c.atm\_location\_id

GROUP BY c.atm\_number, c.atm\_manufacturer, b.location)

WHERE rank <=10;





1. **Number of failed ATM transactions across various card types**

SELECT b.card\_type, COUNT(a.trans\_id) as total\_transaction\_count,

SUM(CASE WHEN a.atm\_status = 'Inactive' THEN 1 ELSE 0 END) inactive\_count,

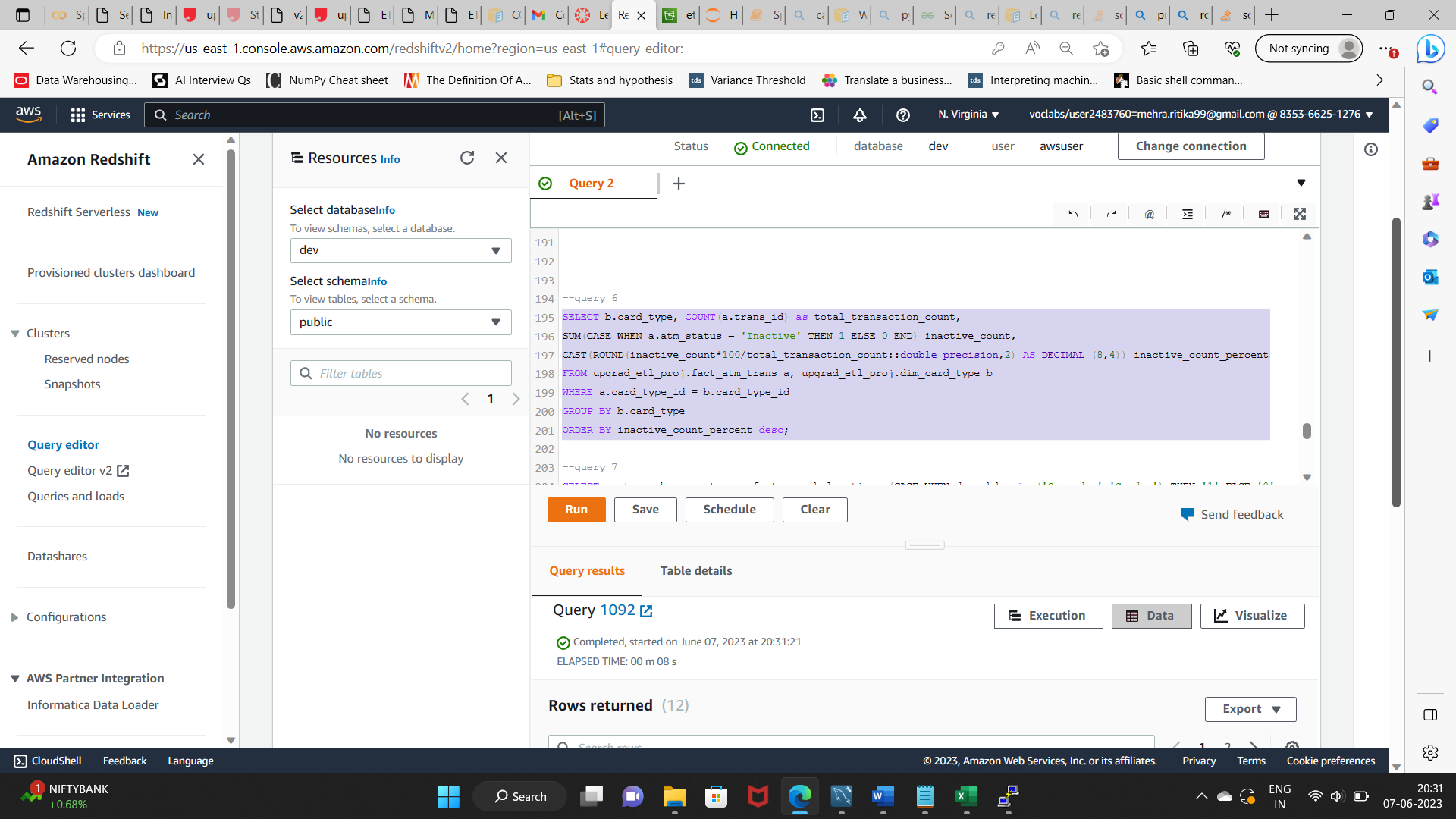
CAST(ROUND(inactive\_count\*100/total\_transaction\_count::double precision,2) AS DECIMAL (8,4)) inactive\_count\_percent

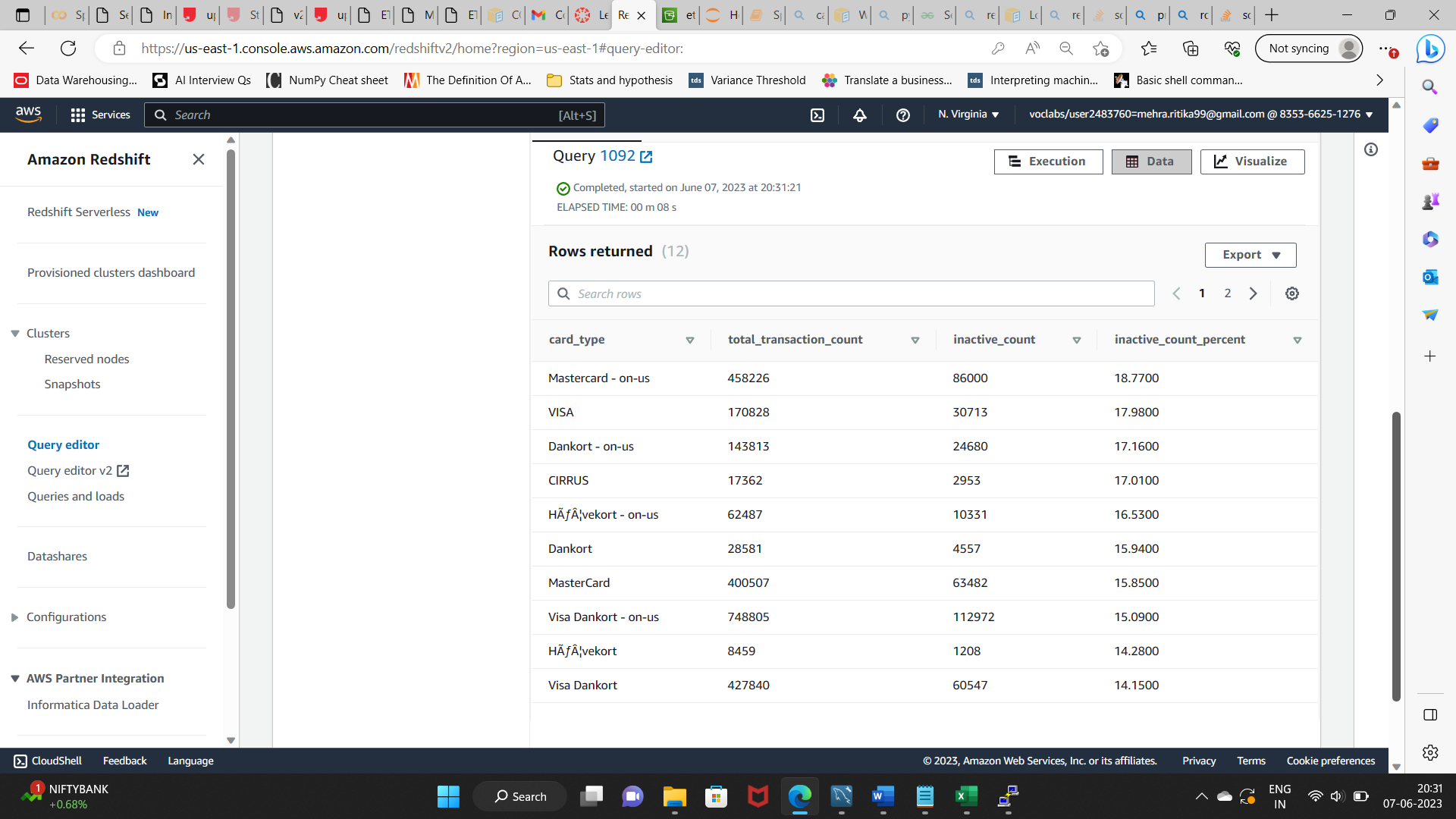
FROM upgrad\_etl\_proj.fact\_atm\_trans a, upgrad\_etl\_proj.dim\_card\_type b

WHERE a.card\_type\_id = b.card\_type\_id

GROUP BY b.card\_type

ORDER BY inactive\_count\_percent desc;





1. **Number of transactions happening on an ATM on weekdays and on weekends throughout the year. Order this by the ATM\_number, ATM\_manufacturer, location, weekend\_flag and then total\_transaction\_count**

SELECT c.atm\_number, c.atm\_manufacturer, b.location, (CASE WHEN d.weekday in ('Saturday','Sunday') THEN '1' ELSE '0' END) weekend\_flag, count(a.trans\_id) total\_transaction\_count

FROM upgrad\_etl\_proj.fact\_atm\_trans a, upgrad\_etl\_proj.dim\_location b, upgrad\_etl\_proj.dim\_atm c, upgrad\_etl\_proj.dim\_date d

WHERE a.weather\_loc\_id = b.location\_id

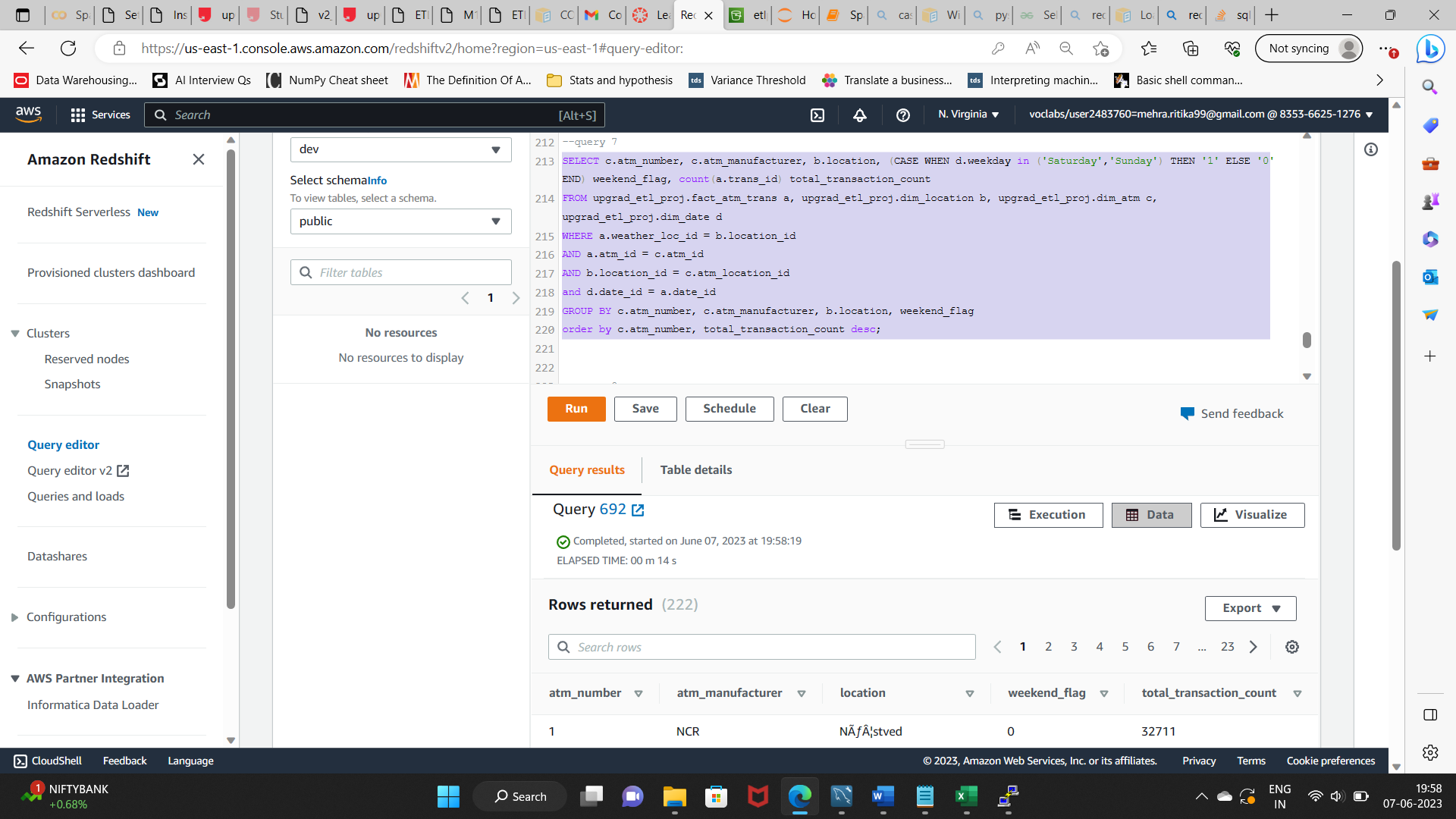
AND a.atm\_id = c.atm\_id

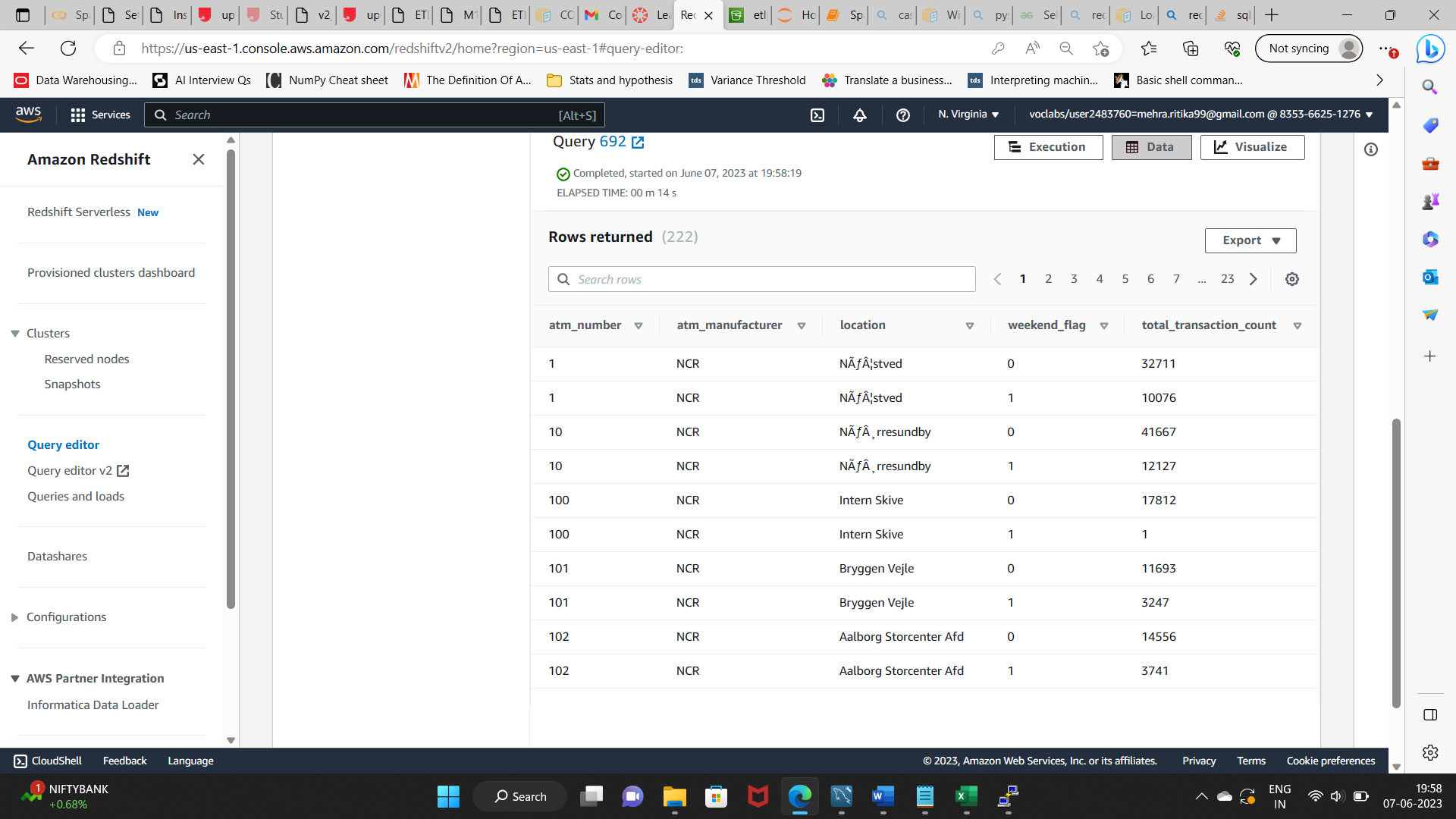
AND b.location\_id = c.atm\_location\_id

AND d.date\_id = a.date\_id

GROUP BY c.atm\_number, c.atm\_manufacturer, b.location, weekend\_flag

ORDER BY c.atm\_number, total\_transaction\_count desc;





1. **Most active day in each ATMs from location "Vejgaard"**

SELECT atm\_number, atm\_manufacturer, location, weekday, total\_transaction\_count from (

SELECT c.atm\_number, c.atm\_manufacturer, b.location, d.weekday, count(a.\*) OVER (PARTITION BY c.atm\_manufacturer, d.weekday) as total\_transaction\_count,

DENSE\_RANK() OVER(PARTITION BY c.atm\_manufacturer ORDER BY count(a.trans\_id) desc) rank

FROM upgrad\_etl\_proj.fact\_atm\_trans a, upgrad\_etl\_proj.dim\_location b, upgrad\_etl\_proj.dim\_atm c, upgrad\_etl\_proj.dim\_date d

WHERE a.weather\_loc\_id = b.location\_id

AND b.location = 'Vejgaard'

AND a.atm\_id = c.atm\_id

AND b.location\_id = c.atm\_location\_id

AND a.date\_id = d.date\_id

GROUP BY c.atm\_number, c.atm\_manufacturer, b.location, d.weekday)

WHERE RANK = 1;

